



Бесплатная электронная книга

УЧУСЬ

recyclerview

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#recyclervie

W

.....	1
<b>1:</b> .....	<b>2</b>
.....	2
Examples.....	2
.....	2
.....	<b>8</b>

---

# Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [recyclerview](#)

It is an unofficial and free recyclerview ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official recyclerview.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# глава 1: Начало работы с переработкой

## замечания

В этом разделе представлен обзор того, что такое `recyclerview`, и почему разработчик может захотеть его использовать.

Следует также упомянуть о любых крупных предметах в рамках `recyclerview` и ссылки на связанные темы. Поскольку документация для `recyclerview` является новой, вам может потребоваться создать начальные версии этих связанных тем.

## Examples

### Установка и настройка

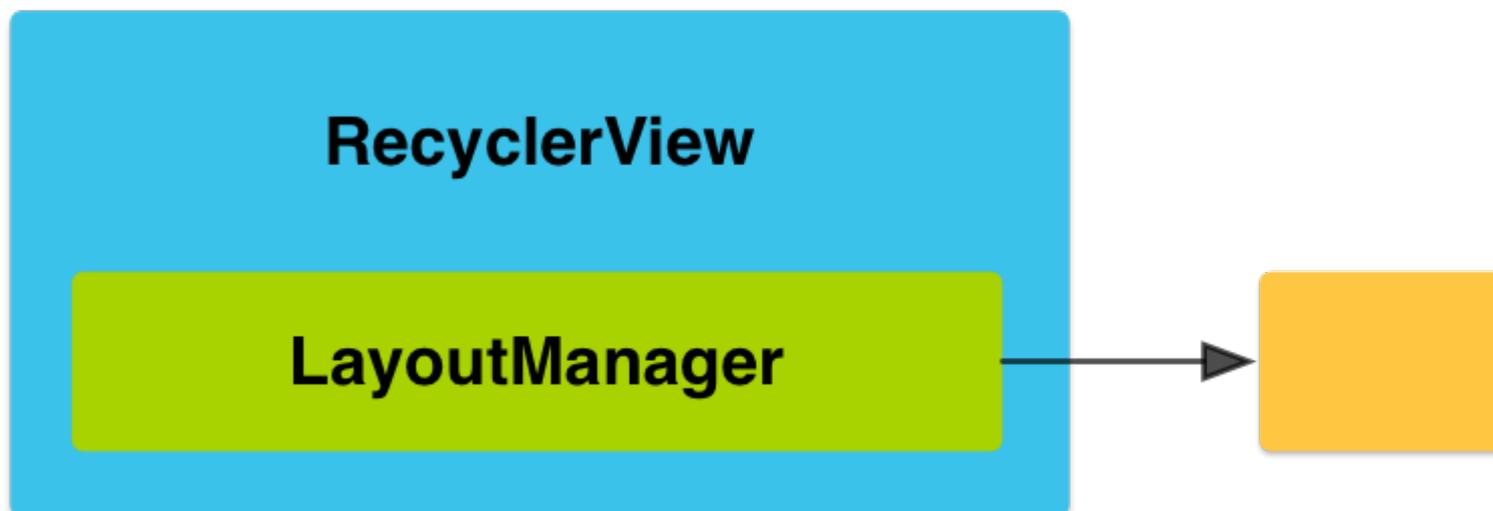
Отображение элементов в списке или сетках является очень распространенным образцом в мобильных приложениях. Пользователь видит коллекцию элементов и может прокручивать их. Коллекция элементов может быть списком, сеткой или другими структурированными представлениями данных.

Виджет `RecyclerView` представляет собой более совершенную и гибкую версию [ListView](#). Этот виджет представляет собой контейнер для отображения больших наборов данных, которые можно прокручивать очень эффективно, поддерживая ограниченное количество просмотров. Используйте виджет `RecyclerView`, когда у вас есть коллекции данных, чьи элементы изменяются во время выполнения на основе действий пользователя или сетевых событий.

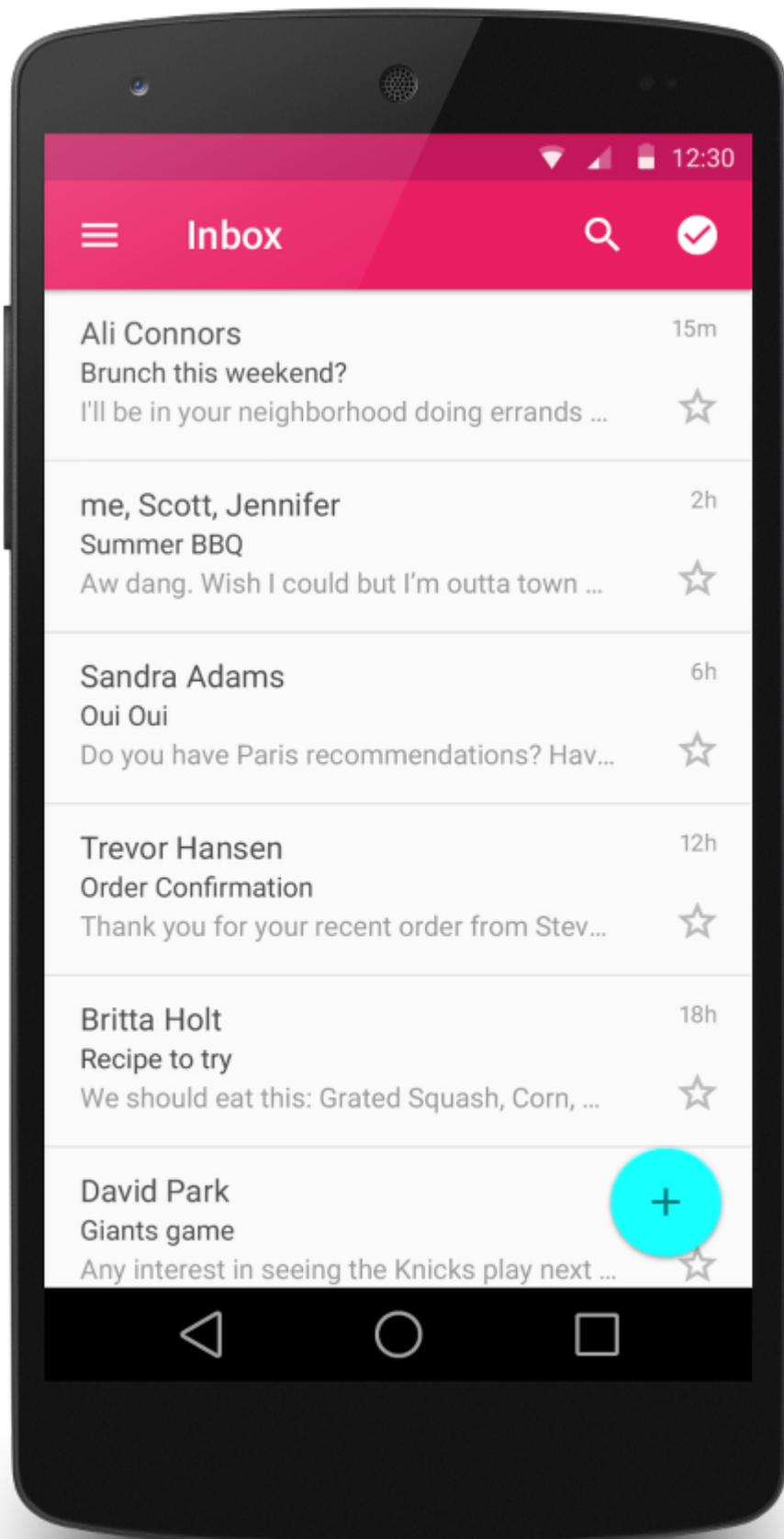
Класс `RecyclerView` упрощает отображение и обработку больших наборов данных, предоставляя:

- Менеджеры макетов для позиционирования элементов
- Анимация по умолчанию для обычных операций с элементами, например удаление или добавление элементов

У вас также есть гибкость для определения менеджеров макетов и анимаций для виджетов `RecyclerView`.



Чтобы использовать виджет `RecyclerView`, вы должны указать адаптер и диспетчер компоновки. Чтобы создать адаптер, расширьте класс `RecyclerView.Adapter`. Детали реализации зависят от специфики вашего набора данных и типа представлений.



Менеджер компоновки позиционирует представления элементов внутри `RecyclerView` и определяет, когда повторно использовать представления элементов, которые больше не видны пользователю. Чтобы повторно использовать (или переработать) представление, менеджер компоновки может попросить адаптер заменить содержимое представления на другой элемент из набора данных. Переработка просмотров таким образом повышает

производительность, избегая создания ненужных представлений или выполняя дорогостоящие поиски `lookViewById ()` .

`RecyclerView` предоставляет эти встроенные менеджеры макетов:

- `LinearLayoutManager` показывает элементы в списке вертикальной или горизонтальной прокрутки.
- `GridLayoutManager` показывает элементы в сетке.
- `StaggeredGridLayoutManager` показывает элементы в шахматном порядке.

Чтобы создать собственный менеджер макетов, расширьте класс `RecyclerView.LayoutManager` .

Теперь, чтобы добавить `RecyclerView` вам нужно добавить следующую зависимость в свой `build.gradle`

```
dependencies {
    compile 'com.android.support:recyclerview-v7:23.4.0'
}
```

Теперь добавьте `RecyclerView` в свой макет, как это.

```
<android.support.v7.widget.RecyclerView
    android:id="@+id/my_recycler_view"
    android:scrollbars="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

Когда вы добавите виджет `RecyclerView` в свой макет, получите дескриптор объекта, подключите его к диспетчеру макетов и прикрепите адаптер для отображаемых данных.

```
public class MyActivity extends Activity {
    private RecyclerView mRecyclerView;
    private RecyclerView.Adapter mAdapter;
    private RecyclerView.LayoutManager mLayoutManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.my_activity);
        mRecyclerView = (RecyclerView) findViewById(R.id.my_recycler_view);

        // use this setting to improve performance if you know that changes
        // in content do not change the layout size of the RecyclerView
        mRecyclerView.setHasFixedSize(true);

        // use a linear layout manager
        mLayoutManager = new LinearLayoutManager(this);
        mRecyclerView.setLayoutManager(mLayoutManager);

        // specify an adapter (see also next example)
        mAdapter = new MyAdapter(myDataset);
        mRecyclerView.setAdapter(mAdapter);
    }
}
```

```
}  
...  
}
```

Адаптер обеспечивает доступ к элементам вашего набора данных, создает представления для элементов и заменяет содержимое некоторых представлений новыми элементами данных, когда исходный элемент больше не отображается. В следующем примере кода показана простая реализация для набора данных, состоящая из массива строк, отображаемых с помощью виджетов `TextView`.

```
public class MyAdapter extends RecyclerView.Adapter<MyAdapter.ViewHolder> {  
    private String[] mDataset;  
  
    // Provide a reference to the views for each data item  
    // Complex data items may need more than one view per item, and  
    // you provide access to all the views for a data item in a view holder  
    public static class ViewHolder extends RecyclerView.ViewHolder {  
        // each data item is just a string in this case  
        public TextView mTextView;  
        public ViewHolder(TextView v) {  
            super(v);  
            mTextView = v;  
        }  
    }  
  
    // Provide a suitable constructor (depends on the kind of dataset)  
    public MyAdapter(String[] myDataset) {  
        mDataset = myDataset;  
    }  
  
    // Create new views (invoked by the layout manager)  
    @Override  
    public MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent,  
                                                int viewType) {  
        // create a new view  
        View v = LayoutInflater.from(parent.getContext())  
            .inflate(R.layout.my_text_view, parent, false);  
        // set the view's size, margins, paddings and layout parameters  
        ...  
        ViewHolder vh = new ViewHolder(v);  
        return vh;  
    }  
  
    // Replace the contents of a view (invoked by the layout manager)  
    @Override  
    public void onBindViewHolder(ViewHolder holder, int position) {  
        // - get element from your dataset at this position  
        // - replace the contents of the view with that element  
        holder.mTextView.setText(mDataset[position]);  
    }  
  
    // Return the size of your dataset (invoked by the layout manager)  
    @Override  
    public int getItemCount() {  
        return mDataset.length;  
    }  
}
```

Сначала добавьте библиотеку поддержки в свой проект: в файле build.gradle приложения добавьте зависимость.

Прочитайте [Начало работы с переработкой онлайн:](#)

<https://riptutorial.com/ru/recyclerview/topic/5426/начало-работы-с-переработкой>

---

## кредиты

S. No	Главы	Contributors
1	Начало работы с переработкой	<a href="#">Community</a> , <a href="#">Princess Ruthie</a> , <a href="#">Reaz Murshed</a>